Comparison of Two Bridge Routing Approaches

Lixia Zhang

Introduction

Both the physical distance that a local area network (LAN) can cover and the number of hosts that can be attached to it are limited. To overcome this limitation, bridges are introduced as devices to connect LANs at the data link layer [1]. The purpose of bridges is to allow hosts attached to different LANs to communicate as if they were on the same LAN. Repeaters, devices that act at the physical layer, allow traffic to cross LAN segments, and all traffic appears on all LAN segments. Bridges, on the other hand, should be more intelligent and should limit traffic to the section of the network on which it is relevant. To accomplish this, bridges must make a routing decision upon each received frame as where to send it to reach its destination(s).

This article compares two routing algorithms proposed for a bridged LAN environment. One is based on creating a spanning tree topology, as introduced in [2]; the other takes a source routing approach, as described in [3]. We identify the features of the running environment and the functional requirements of bridge routing, followed by a discussion of the two different approaches.

Bridge Routing Requirements

In general, LANs are low-cost, low-delay, high-bandwidth (e.g., 1 ~ 10 Mbps) broadcast channels. A bridged LAN environment preserves the low-delay and high-bandwidth features but its topology may be more dynamic than in a single-LAN case due to possible bridge or LAN failures and hosts being moved around.

We consider that bridge routing algorithms should meet the following requirements:

- A bridged LAN environment should resemble a single-LAN environment as closely as possible. In other words, the extension should be transparent to hosts.
- The transparency requirement extends to performance requirements, such as low transmission delays, low undetected data corruption, and keeping frames in order.
- Bridge routing algorithms should be able to adapt quickly to environmental changes.

Features of the Two Approaches

In this section, first, we will describe each of the two approaches briefly, followed by a discussion of the requirements, cost, and functional limitations of each.

Features of the Spanning Tree Algorithm

In this approach, bridges route each frame according to the destination address. To keep frames from looping, a loop-free topology is required. Therefore, bridges must first prune an arbitrarily connected physical topology into a logical spanning tree (ST), such as through running a bridge-to-bridge protocol described in [4].

A bridge listens promiscuously to every frame transmitted on all of its connected LANs. It examines the source address to learn the direction of the source host and keeps a cache table of host-ID/direction pairs (up to a maximum number of pairs as bounded by the hardware). To route a frame, the bridge examines the destination address and looks up its cache. If the destination is found in the cache, the frame is forwarded in the corresponding direction or discarded when the direction is the same as the frame from which it was received; otherwise, it is broadcast in all directions except the one from which it came.

In this approach, bridges are invisible to host stations. No additional host protocol is required beyond that already specified by IEEE 802 for frame data communications across a single LAN. However, the approach does not fully utilize the available resources (because certain channels are disabled to eliminate loops in the topology) and it does not route frames on optimal paths.

The ST scheme places the following requirements on bridges:

1.) Each bridge needs a unique ID that will be used in the spanning tree creation protocol.
2.) Each bridge may need to maintain a large cache of hosts.
3.) Bridges must run at a high speed to keep up with the LAN's data transmission rate.

The total cost of the ST approach is comprised of three parts:

1.) The cost of running the spanning tree protocol—In the absence of topology changes, one control message is transmitted on each LAN with a settable period; during the period immediately

0890-8044/88/0001-0044$01.00©1988 IEEE

after a topology change, more than one control message may be transmitted on a LAN, but the total number is bounded by the number of bridges on that LAN.

2). The cost of topology learning—When a bridge faces an unknown destination address, it takes at least one round-trip to learn the location of the unknown destination (i.e., until receiving a reply from it); meanwhile, the bridge broadcasts all frames addressed to the unknown.

3.) The extra cost of using nonoptimal routes, as compared to using optimal routes.

When topology changes are infrequent, the first two costs are negligible. The last one depends largely on the topology and traffic load. For instance, if the shape of the real topology is close to the logical spanning tree, and if most heavily used server machines are located closely to the root of the tree, the routes being used will be close to optimal ones. On the other hand, if server machines are accidently located at leaves of the tree, the performance may suffer.

The spanning tree protocol provides a number of parameters that users can set to control the shape of the spanning tree [4]. The cost of using nonoptimal routes can be kept low by a careful topology design and proper parameter tuning. Also, although using nonoptimal routes increases the total system load (because the routes are longer), if the traffic is light, as is normally the case in a LAN environment, the effect will be unnoticeable.

One functional limitation of the ST approach is that it does not support multihoming hosts (i.e., hosts attached to more than one LAN). Another limitation is less efficient support for multicasting. The binding between hosts and group addresses may be either static or dynamic. Unless bridges can be informed of such information, multicast frames have to be broadcast through the entire bridged LAN. Finally, the network performance will degrade if the network load ever reaches a high level, and introducing redundant paths will not help because traffic flows only on the single-path spanning tree; the topology redundancy is used only for failure recoveries.

Features of the Source Routing Algorithm

The main issue in a source routing (SR) approach is how routes are discovered. The source routing algorithm proposed in [3] for a bridged LAN works in the following way: the route(s) to a given destination is discovered dynamically by having the source host emit a "discovery" frame, which is then broadcast over the entire bridged LAN. Assuming transmission errors and frame losses are negligible, the route discovery frames will travel through all possible paths between the source and destination hosts; along the way, each frame records the route it takes. When reaching the destination, all the route discovery frames will be returned to the source along the recorded routes. The source can then choose one to use, which will be placed in the header of each frame going to that destination.

Since routes are chosen by source hosts, the bridges' functions are simplified. Upon receiving a frame, a bridge scans the route carried in the header to see if an adjacent pair of LAN numbers matches any two of its attached LANs. If so, the bridge forwards the frame. To avoid duplicate copies forwarded by parallel bridges (i.e., more than one bridge between the same pair of LANs), the algorithm divides the original LAN number field into two parts—the LAN number and parallel bridge number—so that each route can be specified precisely.

Running the SR scheme has the following requirements:

1.) Each LAN needs to be assigned a unique ID.
2.) Each of the parallel bridges between the same pair of LANs needs to be assigned a unique number.
3.) Hosts are required to do route management: finding and selecting routes, monitoring the routes in use, caching newly discovered routes, and deleting obsolete ones.

The cost of running the SR scheme is comprised of three main parts: the cost of the route discovery, the cost of monitoring routes by hosts, and the cost of carrying a full path in each frame's header. If we assume that all LANs have a reasonably large frame size (say one or a few kilobytes), the header size variation under 100 bytes should not raise much concern. In the following, we consider the costs of the first two items only.

Let us look at the cost of route discovery first. Generally speaking, free broadcasting in a mesh topology creates an exponential number of copies and infinite loops. The SR approach eliminates frame looping by two methods: (1) by setting a max-hop limit on each frame and (2) by examining the route each frame has recorded; the frame will not be forwarded to LANs it has already traversed. With the looping eliminations, the number of copies transmitted for one route discovery is approximately on the order of $O(N^M)$, where N is the average number of bridges on each LAN and M the number of LANs in the topology. A computed example is given in the next section.

The cost of monitoring the routes at the host depends on several factors. One factor is the average number of routes being used, which, in turn, depends on the communication patterns of hosts. The cost will be lower if one host talks to few others most of the time. Another factor is the manner in which obsolete routes are detected. One way is to constantly probe each bridge in the path for every cached route, which can be very expensive. Another way is to rely on a higher layer protocol's notification that has failed the route. The latter approach has no detection cost at the data link layer, however, it requires a communication channel across protocol layers to pass the information. This is indeed a good direction for future research, but is not yet available in any current protocol specifications or implementations.

With the required changes to host protocol implementations, the SR approach may provide sufficient data link layer functionalities. Its first limitation is its high cost. The route discovery method can cause the overhead to explode even in a moderate-size LAN complex. There is

[1] *But the final word should be held until the scheme is completely specified.*

no easy way to reduce the cost because the data link layer does not have sufficient knowledge to eliminate the exponential growth of broadcast frames. The SR's mechanism for delivering multicast and broadcast data frames is also expensive because it is the same as the route discovery mechanism. Reference [3] suggests a more efficient mechanism: to have broadcast or multicast frames forwarded only along a precomputed spanning tree. This will cost the same as the ST approach but needs an additional protocol to create the spanning tree first. Some other limitations of the SR approach are discussed in [5].

Comparisons of the Two Approaches

Engineering designs are based on assumptions regarding the environment and judgments between different tradeoffs. To compare the two approaches, we first make estimates on the following parameters in a bridged LAN environment:

- size of a bridged LAN—it may connect a few to a few hundred LANs;
- richness of the topology connectivity—each LAN may be connected to two or more bridges;
- ratio of the number of hosts to the number of bridges—a modest estimate is two orders of magnitude; and
- resource utilization—communication channels are utilized lightly.

Comparison

We will compare the two approaches on the following aspects: transparency, data transmission delay, protocol cost and responsiveness to topology changes, and design strategy.

When considering only changes required to hosts, we see that the ST approach preserves transparency well, while SR requires host protocol modifications for the route management. Because SR sacrifices transparency, it can add features such as packet size discovery along with route discovery. If a bridged LAN contains dissimilar LANs that have different frame sizes, the ST approach must rely on some other means to avoid sending large frames to LANs that have smaller frame sizes.

Once transparency is sacrificed, however, a multitude of other options present themselves for the interconnection of LANs. For instance, any of the popular network layer protocols allow a network of LANs (plus point-to-point links) to be constructed, such that hosts in any part of the network can communicate. Thus, the SR scheme should rather be compared for functionality and performance against the network layer alternatives already available. Other articles in this issue discuss network layer interconnections.

The next aspect of comparison is data transmission delay. At each bridge, the difference in frame forwarding delays between the two approaches depends on how fast the bridge does the host table look-up or the route scanning through the frame header, which presumably is not significant. Along the entire route, the ST method may cause a slightly longer delay in data transmissions on average, due to using nonoptimal paths and the traffic concentration of the spanning tree topology. The SR approach permits optimal routing, but hosts will suffer a connection start-up delay if no path to the destination has been previously cached. In addition to delay, route discovery is also expensive, hence the question of how long to keep unused routes in cache becomes an issue of much concern, which will be discussed next.

The total overhead of the ST approach goes up linearly with the number of LANs, but keeps constant on each LAN. The overhead of the SR approach grows exponentially with the diameter of the topology; it also goes up with the total number of hosts since each host must perform its own route discoveries. One way to cut the cost is to let hosts save the route after a connection is closed. To decide how long routes can be kept, however, requires knowledge about the dynamics of the network environment, such as how often hosts go off-line or change locations, how often bridges or LANs fail, etc. Coding such knowledge into hosts sounds infeasible, because any later changes will require a change to all hosts.

To see how quickly the ST can react to topology changes, we have divided the changes into two types: host location changes (including going off-line and coming on-line) and network topology changes (i.e., the bridge or LAN going up or down). Hosts can announce their changes after moving or coming on-line to let bridges learn the new locations immediately. The recovery time from a topology change is proportional to the topology diameter.

In the SR approach, the response time to topology changes is up to the route caching policies. If hosts perform route discovery for every new connection and rely on the higher layer protocols to detect topology changes, the problem becomes how fast an active connection can detect a route failure. If routes are cached independently from connections, then comes the question of how to remove obsolete routes promptly from the cache. Decisions on route caching policies and algorithms are yet to be made, and the analysis must wait until the algorithms are completely specified.

Some investigation on the design strategies of the two approaches may help us further understand their differences. Both designs assumed that bandwidths are inexpensive in a bridged LAN. The ST approach exploits nonoptimal but relatively simple routing strategies which does not require host changes, and uses the assumption to justify the use of nonoptimal routes. In contrast, the SR approach uses the assumption to justify an expensive route discovery protocol, which then finds optimal routes for data transmissions.

Another design difference is functionality versus performance. The ST was designed to be as simple as possible, so long as it could accomplish the mandatory task of delivering frames in a bridged LAN; it assumes that the sufficient resources in a bridged LAN will automatically provide good performance. While motivated by achieving good performance, the SR was built with a much richer set of functionalities, such as finding optimal paths, being able to use parallel paths, etc. With a light traffic load, however, the cost of finding optimal paths may not be justified by the small saving from using those paths. Also, finding parallel paths is of little use if the data link layer does not have the knowledge to make a good selection. For instance, splitting load to multiple paths requires knowing the desired throughput value; otherwise, the data link layer cannot decide whether more than one route will be needed. Also, if the data link layer does utilize multiple paths, it must have some means of putting packets back in order, or it must be sure that higher layer protocols have assumed that the data link layer will reorder packets.

The two designs also differ as to where to install the routing functionalities—at the bridge or at the host. Because the topology changes dynamically, the routing agent also needs to do constant checking in order to promptly correct broken routes. The ST dedicates the task to bridges; the SR to hosts. The number of hosts in a LAN complex is usually two orders of magnitude higher than the number of bridges. It is easier and less expensive to have bridges rather than hosts keep track of the network status.

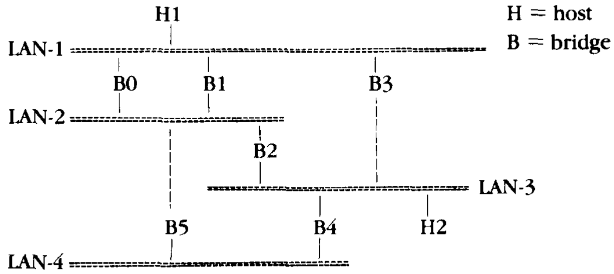
In general, it is desirable to build network functions into switches to release the burden from hosts; it is also desirable to simplify bridge designs to achieve the high throughput required by a high-bandwidth LAN environment. The ST design aims at meeting both by simplifying the routing functions in bridges; the SR design sacrifices one of the two, and simplifies bridges by shifting the routing functions into hosts.

Although not all differences between the two approaches have been mentioned in this paper due to space limits, a few comments about source routing in general are worth mentioning. Much work has been done in the direction of source routing [6-8]. Briefly, source routing gives hosts a control over communication paths, therefore, it can provide hosts with flexible routing functionalities and controllabilities. It also helps to resolve the addressing difficulties in a confederation of networks with heterogeneous address structures, because the route can be used as addresses in certain cases. In order not to burden hosts with the responsibility of keeping track of network dynamic status and route handling, routing servers have been proposed to take over the responsibility.

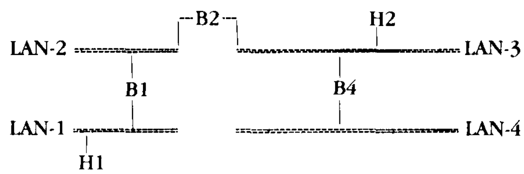
In general, hosts desire controllability when services by different communication paths make a difference, for instance, in terms of price, quality, administrative restrictions, etc., the considerations that normally do not occur in a transparent LAN environment. Although a bridged LAN may connect LANs of different types, it is still a rather homogeneous environment, as compared to an interconnection of long-haul networks. The latter generally connects networks under multiple administrative agencies, and the capacity differences among component networks are often as much as several orders of magnitude; while a bridged LAN normally is assumed to be confined to private premises, and the difference in the channel capacities is within an order of magnitude. Therefore, the many functionalities provided by the SR approach do not seem very useful.

Example

In order to give a clearer understanding of the relative costs of the two approaches, we compute the costs of the two in the following example.



Let us first compute the cost of the ST approach. After *N* Hello message exchanges, where *N* is on the order of the network diameter [4], the topology will be pruned to the following spanning tree:



Assuming that H1 starts communicating with H2, and no bridge has cached any information about H2 by then, the frames from H1 addressed to H2 will be forwarded by all of B1, B2, and B4. Frames reach the destination on LAN-3 but B4 fowards frames unnecessarily onto LAN-4. This unnecessary traffic on LAN-4 will persist until B4 receives a frame originated by H2.

Now let us look at the cost of the SR approach for the same case. First, H1 sends a route discovery frame on LAN-1. When B0, B1, and B3 receive the frame, each of them will try to forward it further to LANs it has not passed through. The original route discovery frame will then be fabricated to multiple copies on other LANs (the exact numbers are: four will pass on LAN-2, five on LAN-3, and six on LAN-4), approximately a factorial number of the LANs in total, since each copy tends to travel through all LANs. Correspondingly, all bridges will also have to process multiple route discovery frames (from four to six copies in this case) generated by just one request. When five route discovery frames following the five distinct routes finally arrive at H2, they will all return to H1 again, creating additional cost.

Note that the preceding example is of a very small bridged LAN, consisting of only four LANs. With a more realistic size network, consisting of, say, 12 LANs (still quite a small size), and with about the same connectivity as the preceding example, the number of frames to arrive on each LAN with the SR scheme would approach the order of 2^{12}, or 4,000, whereas overhead from the ST scheme would not grow significantly.

Summary

Different protocols and algorithms are designed for different purposes and different network environments. The superiority of two algorithms depends heavily on the assumptions of their running environments.

The source routing approach, as proposed in [3], provides more functionality (e.g., multiple paths, optimal routes, maximum packet size discovery) along with a higher cost and requirements of changes to host implementations. It also has several unresolved technical issues (such as the route caching strategy). Generally speaking, source routing can provide more functionalities; they are rarely needed, however, by the data link layer in a transparent local area network (LAN) connection. Putting the responsibility of route handling into hosts is also not well justified.

Although some extra cost is paid through routing frames on nonoptimal paths, the spanning tree approach itself is inexpensive and very simple. It is a better choice for light-traffic LAN environments with a properly engineered topology.

References

[1] Zimmerman, H., "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. Commun.*, Vol. 28, No. 4, pp. 425-432.

[2] Backes, F., "Transparent Bridges for Interconnection of IEEE 802 LANs," *IEEE Network* this issue.

[3] Dixon, R., and Pitt, D., "Addressing, Bridging, and Source Routing," *IEEE Network,* this issue.

[4] Perlman, R., "An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN," *Proc. 9th Data Commun. Symp.* 1985.

[5] Hart, J., "Interoperability Between 802.5 and 802.1 Bridging Techniques," unpublished paper. The author is with Vitalink Communications Corp. Oct. 1987.

[6] Farber, D., and Vittal, J., "Extendability Considerations in the Design of the Distributed Computer Systems," Proc. Nat. Telecommun. Conf. 1973.

[7] Sunshine, C., "Source Routing in Computer Networks," *Computer Commun. Rev.* pp. 29-33, Jan. 1977.

[8] Saltzer, J., Reed, D., and Clark, D., "Source Routing for Campus-Wide Internet Transport," *Local Networks for Computer Communications,* IFIP, 1981.

Lixia Zhang received the B.S. degree in Physics from Heilongjiang University, China, in 1976, and the M.S. degree in Electrical Engineering from California State University, Los Angeles, in 1981. She is currently working toward her Ph.D. degree at the Massachusetts Institute of Technology. Her research interests include computer network architectures and protocols, and distributed systems.